Data Gathering

This project contains data gathered from three dataframes:

- twitter_archive
- image_predictions
- tweet_counts

The first dataframe, twitter_archive, was obtained by downloading the .csv from course resources.

The second dataframe, image_predictions, was obtained by accessing the .tsv using the requests library, then opened and read into an image_predictions.tsv file.

The third dataframe, tweet_counts, was obtained using the twitter API. I used the tweet_ids from the twitter_archive to specify which tweets to additional information and saved it in a dictionary. Then, parsed the tweet_id, favorite_count, and retweet_count from each line of the of the dictionary to create the tweet_count dataframe.

Assessing Data

Twitter Archive

To assist with understanding the columns, I found a Data Dictionary from the Twitter Developer Platform.

In assessing the data I found…

- Some of the datatypes were not stored incorrectly such as tweet_id and timestamp — using .info( )
- There were only 4 unique source types — using visual assessment & .unique( )
- Lowercase entries in the names column weren't actually dog names — using visual assessment, then .str.islower( )
- All observations were prior to August 2017 — used .str.contains( ) with specific years
- There were some observations with missing pictures, missing ratings, holiday related ratings and incorrectly stored ratings — initial concern was discovered from.describe() and seeing very high and low rating_numerators and rating_denominators. I decided to look into the tweet text and images for a few select observations that had high rating_numerators, high rating_denominators, and low_rating denominators.
- A few of the columns could be combined into one to fix a tidiness issue - using visual assessment.

Image Predictions

Upon both visual and programmatic assessment, the only issue I found was a datatype error with the tweet_id.

Tweet Counts

In assessing the data I found…

- tweet_id was stored incorrectly - using .info( )
- Some of the observations with zero favorite_counts were retweets or didn't have an image — using .describe( ) then looking into the tweet text & images of a few select observations

Tidiness

- Since each observation was present across all dataframes, all three of them should be merged into one.

- Since this project isn’t interested in retweets & replies, those observations should be dropped, along with columns that deal with retweets/replies.
- Some of the columns can be merged into one since they describe one variable, dog nicknames.

Cleaning

Merging Dataframes

- Merged all three dataframes into one based on the tweet_id.

Merged Dog Nicknames

- Wrote and applied a function to consolidate the 4 dog nickname columns into one “nickname” column.

Dropped Irrelevant Information

- Dropped Retweets by generating list of the observations with notna values in the retweeted_status_id column.
- Dropped Replies by generating list of the observations with notna values in the in_reply_to_status_id column.
- Dropped a tweet a non-dog photo by calling upon the index based on the tweet_id.
- Dropped irrelevant columns that didn’t pertain to the project:
    - “in_reply_to_status_id"
    - “in_reply_to_user_id"
    - “retweeted_status_id"
    - “retweeted_status_user_id"
    - “retweeted_status_timestamp"
    - “doggo"
    - “floofer"
    - “pupper"
    - “puppo"

Convert Datatypes

- Converted timestamps from string to datetime.
- Converted tweet_ids from integer to string.

Source Column

- Extracted the text from the source url to output just the source name.

Fixed Entries

- Wrote and applied a function to replace lowercase entries in name column with “None”.
- Replaced the rating_numerator and rating_denominator for a specific tweet_id with the correct information.

Rating Column

- Created a rating column as a ratio of rating_numerator and rating_denominator to make analyzing the data easier.